Tele-service through the InterNet-world-wide-web of laboratory equipments according to the NAMUR recommendations and remote control by an integrated camera

Baez Galeano.Julissa M, Rabesiaka.Rivo J, Razanamparany. Bruno, Razafindrazanakolona. Daniel, Andrianantenaina.M Hajamalala

Abstract— **Research laboratories and the current enterprises turn toward the automation of their instruments [1]. Not having not a robust and versatile enough control software can become disadvantageous [2] and will involve a supplementary load for the purchase and the staff' straining. Hence, this work is the fruit of the development of a control interface through Internet of the equipments that follows the NAMUR recommendations. It permits to remote control and to integrate easily new equipments due to the flexibility and to the conviviality of its interface without needing computer literacy. Some applications were carried out to measure the hardiness and the reliability of this new system. In spite of the few observed mistakes, one can conclude that the developed software is reliable. This application satisfies the primordial needs of the user: adaptation capacity and easiness of use [3]. It also has an attractive external functional capacity following the 9126 norm that defines the quality of a software [4,5].**

Index Terms— **Software, customer - server, automation, Internet, remote control, NAMUR recommendations.**

I. INTRODUCTION

Before the phase of development, the automation of the processes constitutes an important step and aims at optimizing the processes of production [1]. Until this days, the processes has been controlled by hand but it tends more and more towards automation and the programming of the instruments [1]. This is how the tele-service through Internet is made possible. The tele- service by Internet permits to control and to follow the evolution of the values of the one or the several parameters values sent by the of laboratory equipments from a distance.

Knowing the present technological advancement, there is a permanent innovation of these laboratory facilities. Therefore, the user will not have the possibility to integrate these new devices coming from different constructors because of the incompatibility of the software required by the machines. The introduction of new equipments often requires the renewal of the control Software which would entail a supplementary cost for the purchase and the training of the users [1].

The man - machine interface is therefore a vital factor in the world of automation and must satisfy the needs of the currently more and more complex processes [6]. This work permits to develop an interface to control and to supervise remote laboratory equipments according to the NAMUR recommendations. The NAMUR recommendations aims at normalizing the instructions related to males - females' connect that are necessary to the transmission of analogical and numeric signals intended to control laboratory processes [7]. A lot of equipments of different constructors follow these recommendationsThe competition and the fluctuation of the market in the domain of the software conception keep increasing. Therefore, one must take into account the quality of the software starting from its conception in order to give satisfaction to the users [8].In fact, the available Software on the market have an opened architecture to allow the users to use all manufacture applications within an automation system [1,9]. The use of the Software often requires a computer literacy because of their object oriented structures. However, the user is not obliged to have a deep knowledge in programming. This leads to a difficulty of adaptation to such structure and can require a need of training for its users [9].

The case is different for our application. Actually, to exchange information between the computer and the devices, the computer must send some instructions to the equipments. These instructions change depending on the type of equipment and on the constructor. To get this possibility to command and to control from a distance all the equipments, two applications were conceived: a customer application and a server application. This last permits to record the data of the equipments in the register of Windows. It is the same for the objects that constitute the server interface: the operator can modify the name of the objects and record them to be consistent with their use. These data are then available at the opening of the application and the operator would have the possibility to command the equipments from distance due to a convivial interface and a simplified use.

The conceived software permits to integrate efficiently these materials in the process of automation covering several domains of the science such as the chemistry, the food engineering, the biology, the physics…

All necessary tools to develop a of complete and powerful automation application as well as all required elements for a professional solution of supervision and control that are provided in the application: data acquirement, visualization, centralization and distribution of the information, man - local machine interface, distant web or Intranet access, evolutionary interface and possibility of integrating all laboratory equipments from various constructors according to the NAMUR recommendations, remote control by a camera. Note that the software was conceived according to the ISO 9126 standard [10].

This article aims at presenting the conception of these two software which will allow to command from a distance all laboratory equipments according to the NAMUR recommendations. The server permits to record the instructions that command these equipments in the register of Windows. The client software communicates from a distance with the server in order to exchange information with the equipments listed in the register of Windows.

II. Material and Methods

A. Material

Numerous equipments that follow this NAMUR recommendation are connected on the asynchronous links RS-232 of a computer.

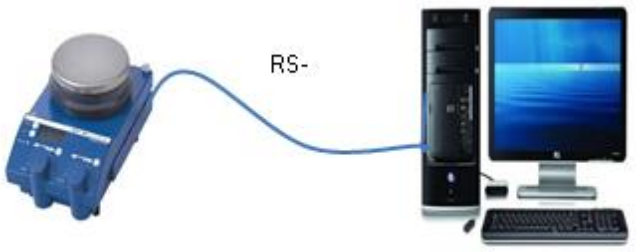

Figure I: Link between the computer and the equipments according to the NAMUR instructions.

Physical links alone do not permit to communicate with these equipments. The Man - machine interface that is the command platform is necessary [11].
This Software joins two distant computers by Internet link, and locally this software joins the equipments and the computer by cable

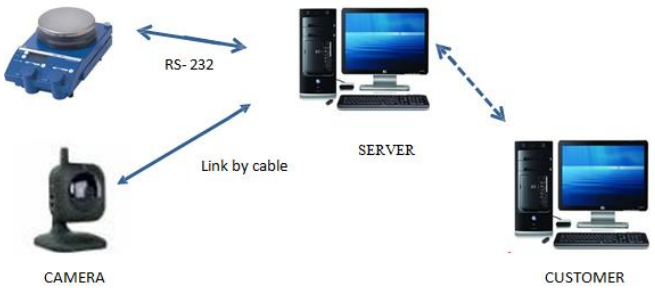

Figure II: Remote connection with the equipments.

The Internet link permits to have the communication of the functionality from a distance. This communication between two computers presents an analogy with two programs that communicate for which the technology is based on the customer - server application [12].
The customer interface sends commands to the server interface while this latter receives the information and transmits it to the laboratory equipments through the link by cable.
Thus, at the equipment level, the following devices are required:
- A client computer and a server computer.

An equipment that follows the NAMUR recommendations for validation. We used a heating agitator IKA RET control visc.

Actually, this equipment possesses a RS-232 interface enabling to control it according to the NAMUR instructions. It presents several functions: agitation, heating and viscosity. The available instructions [7] are summarized on the table 1.
- A camera: the camera permits to follow the progress of the operation from a distance in real time

NAMUR Instructions	Function
IN_NAME	Designation demand
IN_PV_X X=1;2;3;4;5; 7; 8;	Reading of Real value
IN_SP_X X=1;2;3;4;7;8; 12;42;50;52;53;	Reading of the selected theoretical value
IN_TYPE	Demand of the laboratory device detection
OUT_SP_X n X=1;2;4;7;50;52,…,5	Selection of a theoretical value equals to n
RESET	Deactivation of the device function.
START_X…	Activate the link between the material and the computer

Table 1: Some instructions of agitator IKA.

As far as the Software is concerned, two command software are required: a client software which manages the client computer and a server Software which manages the server computer.
Our objective is to develop these two platforms while giving this capacity of adaptation to communicate from a distance with all laboratory equipments that follow the NAMUR recommendations via the physical link by cable and via Internet.

B. Methods

1) Principle

Two applications were created: a customer application and a server application for which the conception is presented by the diagram below:

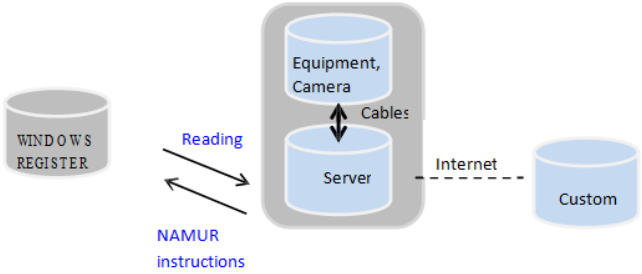

Figure III: Conception of the customer and server applications.

The server application ensures the local communication with the laboratory equipments and the camera. The customer application permits to communicate with the equipments via Internet. The customer can send some information to the server computer (instructions about temperature or agitation for example) and at the same time to receive the data coming

from this server: reading of the real temperature values, agitation or relative viscosity,...
The server interface permits to reach these data and to command all equipments that follow the NAMUR recommendations from a distance and that are listed in the register of Windows without needing any preliminary knowledge in Data processing. The structure of the files and the keys of the register are summarized in the figure IV:

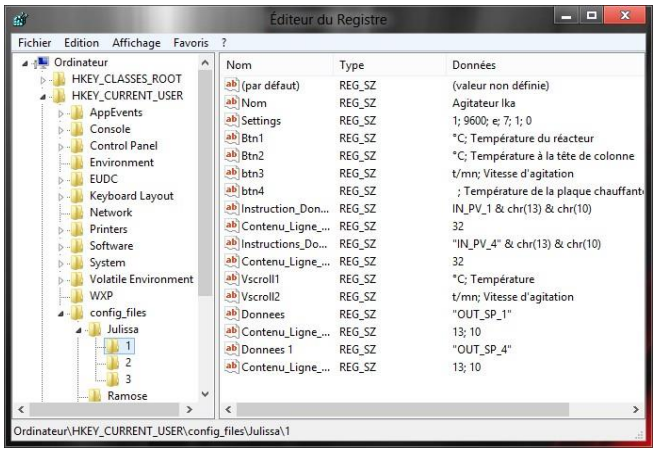

Figure IV: Structure of the files and Data recorded in the register of Windows.

The developed customer - server interfaces were conceived by using the Visual Basic 6 language. These interfaces join the 2 computers through Internet link. All genius shop softwares can correctly manage all the sets of relative functions to the networks and especially to the Internet. This Visual Basic language presents an object oriented approach [12] and permits to quickly develop an application due to the possibility to manipulate some objects in
order to reuse them and to optimize the writing and the speed of the code. Considering all integration circumstances of several techniques of communication (use of the serial port, the Internet, the Windows register and possibly the USB port), Visual Basic 6 was chosen to conceive quickly and with high flexibility the applications under Windows and to access the communications of the computer ports as well as the functions for connection networks.

2) Methods of Programming

Visual Basic 6 allows to manage an application while developing a convivial interface to communicate between humans and equipments. The Visual Basic 6 work environment is often designated under the name "integrated development Environment": it integrates numerous and varied functions such as the creation, the modification, the compilation and the debugging within a same environment [12]. The creation mode serves to conceive the application and as this language presents an object oriented approach [12], the components in Visual Basic are considered as objects. These objects possess their properties, methods and events. The properties are objects attributes (dimension, color, shape, name…), the methods are comparable to actions and events answers.
The programming has 3 main stages:

1 - Development of links with the computer ports: the "Mscomm" component is an object of Visual Basic. This component is destined to put the links in place with the serial port (RS-
232) of the computer due to the different constants and properties allocated to this port. As any object of Visual Basic, this component possesses its own properties, methods and events.
The properties of Mscomm (table II) permit to especially configurate the ports, to send and to receive some data on this port.

Proprerties	Description
Commport	Defines and sends back the port number.
EOFEnabled	Searches for the end of file during the
Handshaking	Defines and sends back the protocol of
InbufferSize	Defines and sends back the size of the
InputMode	Defines and sends back the type of data excerpt by the property Input
InputLen	Defines and sends back the number of characters read in the tampon of receipt
OutbufferSize	Defines and sends back the size of the transmission tampon in bytes.
PortOpen	Open the port.
Settings	Defines and send back the parameters of the port (speed, parity,…)

TableII: Some interesting properties of the component MsComm[16]

The MsComm control possesses only one event "On_Comm" that is triggered when an event or a mistake of communication occurred. This "On_Comm" event characterized by several constants that describe the state of the "MsComm" component at the time of its use. We are going to present an example of lines of commands to use this "MsComm" component and to sort the data coming from the receipt tampon of the equipment:

```
Private Sub Timer4_Timer()
Dim x as string
Dim stock As String
Dim pas As Integer
Dim longeur as Integer
MSComm(i).RTSEnable = Label_Flux
MSComm(i).Output = Instructions_Donnees
x = MSComm(i).Input
pas = 1
longeur = Len(x)
stock = Mid(x, 1, 1
If stock = Contenu_Ligne_LL Then GoTo Line1 Else GoTo Line2
Line1
stock = Mid(x, pas + 1, 1)
Goto 10
Line2:
10  If Mid(x, pas + 1, 1) = Contenu_Ligne_LL Then GoTo 12 Else GoTo line4
Line3:
```

```
pas = pas + 1
If pas <longeurThenGoTo 11 ElseMsgBox "Erreur"
Line4:
stock = stock & Mid(x, pas + 1, 1)
pas = pas + 1
If pas <longeurThen
GoTo 11
Else
10 Text16.Text = stock
End If
If Text16.Text = "" Then Text6.Text= Text6.Text
'Text6.Text =
Mid(Text6.Text, 1, 4)
Else
Text6.Text = Text16.Text
End If
End Sub
```

2- Setting up of the link between the 2 programs that communicate via the Internet (customer - server application): we created the sockets as well as the pertaining methods. The sockets are objects of Visual Basic. They manage the communications by the TCP/IP protocol between two programs via 2 distant computers. And like any object of Visual Basic, the sockets also possess their properties, methods and events.
The principle of this link is to communicate by link active X to dll named "wsock32.dll".

```
Function (Buffer As String)
Dim cr as Long
Const MAX_BUFF_SIZE = 10000
Dim buff(0 To MAX_BUFF_SIZE) As Byte
cr = recv(CurrSocket, buff(0), MAX_BUFF_SIZE, 0)
If cr< 0 Then _
GetErrorDescriptionListIndex, WSAGetLastError(),
"ReceiveData"
buff(cr) = 0
Buffer = Left(StrConv(buff(), vbUnicode), cr)
End Function
```

These functions can be called at any time. Here is an example of a code that permits to recover data in the receipt tampon such as calling the created GetData method:

```
Function Winsock_ReceiveData(Index As Integer)
Dim message as String
Winsock.GetData message
Text18.Text = message
End Function
```

The sockets constitute a simple and quick approach to construct real application networks combined with the graphic environment of Windows with a written V.B 6 application [12].

3 - Writing and reading of the data concerning the equipments in the register of Windows.
The Register contains some information concerning the operating system and the applications sheltered on the computer. The access to the register uses the integrated functions of Visual Basic as [12]: DeleteSetting, GetAllSettings, GetSetting, and SaveSetting, and also the Registry classes and RegistryKey of CLR (Common LanguageRuntime).

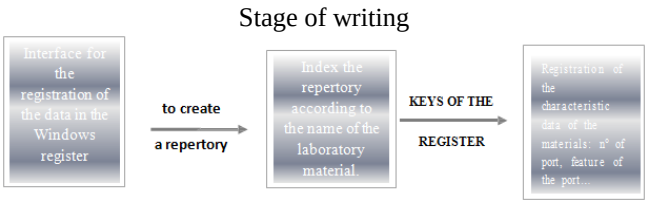

Figure V: Stage of writing on the Windows register.

b- Stage of reading

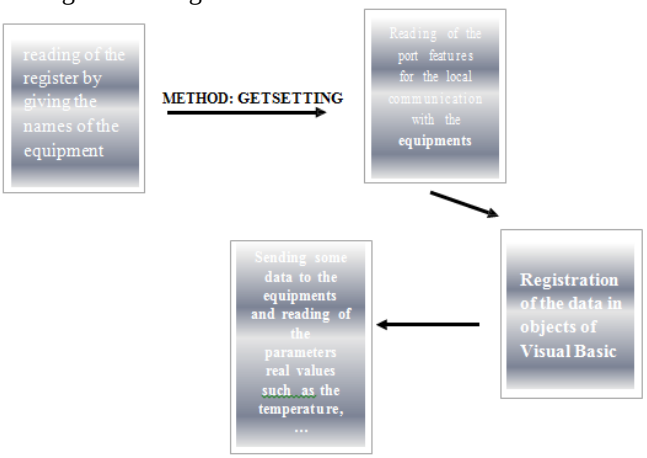

Figure VI: Stage of reading on the register

Here is an example of reading and writing on the register by calling the " Read "and" Write functions "that we created:

```
Parite = List_Parite
If Text_Donnees_Autre.Text<> "" Then
If IsNumeric(Text_Donnees_Autre.Text) = False Then
reponse = MsgBox("Vous devez entrer un chiffre",
vbCritical, "Nombre de bits de données - Erreur de saisie " )
GoTo Line1

End If
Bits_Donnees = List_Donnees
If Text_Stop_Autre.Text<> "" Then
If IsNumeric(Text_Stop_Autre.Text) = False Then
reponse = MsgBox("Vous devez entrer un chiffre",
vbCritical, "Nombre de bits de stop – Erreur
Goto Line1
End If
Else
End If
Bits_Stop = Text_Stop_Autre.Text Bits_Stop =List_Stop
Call Ecrire(CStr(Chemin_registre) +"\"+
CStr(Numero_Module_Actif), 12, Parite)
Call Ecrire(CStr(Chemin_registre) + "\" +
CStr(Numero_Module_Actif), 13,
CStr(Nombre_Bits_Donnees)) Call
Ecrire(CStr(Chemin_registre) + "\" +
CStr(Numero_Module_Actif), 14, CStr(Bits_Stop))
Parametres.Height = 2090
Call Lire(CStr(Chemin_registre) + "\" +
CStr(Numero_Module_Actif), 12) Parite
= Contenu_Ligne_LL
Label_Parite.Caption = Parite
```

```
Call Lire(CStr(Chemin_registre) + "\" + CStr(Numero_Module_Actif), 13) Bits_Donnees = Contenu_Ligne_LL
Label_Donnees.Caption = Bits_Donnees

Call Lire(CStr(Chemin_registre) + "\" + CStr(Numero_Module_Actif), 14) Bits_Stop = Contenu_Ligne_LL
Label_Stop.Caption = Bits_Stop
```

III. Results

A. *Presentation of the customer interface*

The customer interface includes one main sheet in order to simplify its use and to follow the second point required by the ISO 9126 norm of quality thus:

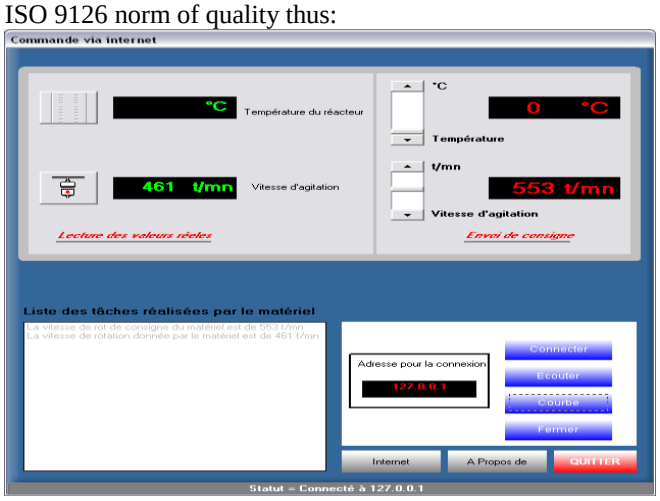

Figure VII: customer Interface

B. *Presentation of the server interface*

The server interface permits to command the heating and agitation functions of an IKA ret Control visc heating agitator:

Figure VIII: Main sheet of the server interface

The statute "Non Connecté" indicates that one is not connected to a client computer. The statute of the connection is presented in this tab.

This interface presents the menu which also permits to change the equipments to be commanded as well as the name of the objects that constitutes the server interface. One can record the instructions for these equipments in the register of Windows. The operator can also modify the name of the objects of the interface (the buttons orders, the units of measures…) and to record them so that they are in conformity with their use. The recorded equipments can be commanded from a distance by the client software:

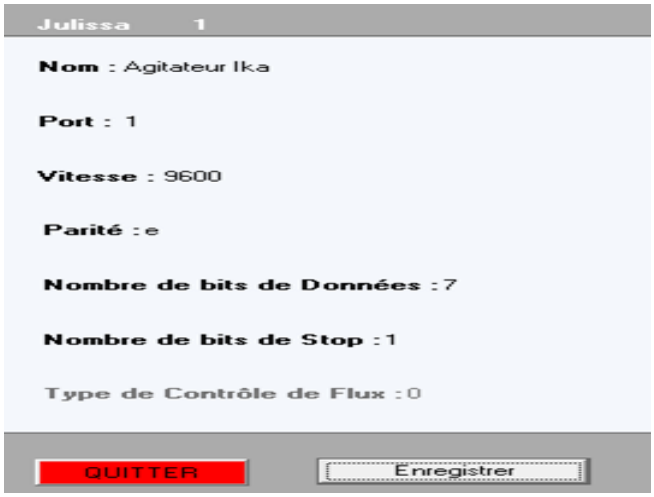

Figure IX: Data registration in the register of Windows

IV. Discussion

A lot of software automations turn to the use of an object oriented language as the C++, Java or the VB.net, to have an application opening of the. It is for example the case for Berman, R.T [1]; and for the SIEMENS software platforms [2]. These platforms provide some mechanisms to develop reusable functions in order to optimize the writing and the speed of the code.

In the world of laboratory automation, the software flexibility is primordial. This flexibility results in the limitation or no of a software use [8]. Indeed, it is advised from its conception to think about the possibility to control a system of any dimension [3], that means a system constituted of several equipments even from different constructors, of a camera.... that reflects our case. The remote control is also important and gives more flexibility to such structure. Actually, this service permits to follow a process without needing to be there. In this case, the camera assures the progress of an operation from a distance in real time. The modern laboratories tend to use these approaches in order to adopt more suitable securities measures and to guarantee the security of an installation [9].

An object oriented language possesses 3 important programming mechanisms: the encapsulation, the inheritance and the polymorphism. The encapsulation permits to unite the variables and functions within the same entity named class [1]. The inheritance is a technique to define a class hierarchy. Every daughter class inherits the methods and the data of its "mothers". In addition, the polymorphism offers the possibility to define several functions of the same name but different parameters properties [1]. The set of these 3 tools allows to develop a complete library of objects that can be reused [1].

The opening of an architecture permits to take these objects (functions, methods…) from another programming language that presents this object oriented approach and give this flexibility to the automation of software package. To use this software, it is necessary to choose the language and eventually to learn it. In fact, such software structure presents a good

portability but often requires an adaptation period for the users and a preliminary deep knowledge in language programming. That is not the case for our application.

We want to signal that, at the time of the software conception, we followed the ISO 9126 norm [13,2,10], a standard to measure the software quality, at the time of its conception. This norm requires a certain attractive capacity use and qualifies 6 groups of quality indicators for the software such as:

- the functional capacity, that evokes the software functionalities to answer the explicit or implicit needs of the users, the easiness of use, concerning the necessary effort to learn and to manipulate the software,
- the reliability and the hardiness, that means the capacity of a software to provide correct results under any kinds of exploitation conditions ,
- the performance: that is the yield related to the quantity of resources used (material possibilities, time, personal), and the quantity of results delivered,
- the maintainability, concerning the easiness or not to correct and to transform the software,
- the portability, that means the software faculty to function within an equipment or a software environment different from its initial environment.

On one hand, due to an intuitive interface our software package permits, to introduce the necessary instructions for the control – command of these equipments. The end user will have the possibility to command easily and even from a distance the equipments that follow the NAMUR recommendation without knowing programming.

In fact, automation software must have these new perspectives concerning flexibility and maximum efficiency. In spite of unceasing apparition of new ranges of equipments on the market, it is possible to integrate these novelties due to the interface flexibility.

On the other hand, hardiness is also a characteristic that must be taken into account at the time of a software conception [2,3]. It is shown by its capacity to function in an unforeseeable or non valid situation [14]. In fact, some experimental tests were carried out to ascertain the hardiness of this application:

- Curve of temperature and agitation speed remote control against time. In spite of unforeseeable exploitation conditions at the time of the experimentation due to the relatively low debit of the local Internet, some mistakes were observed but they did not have any impact on the results. This reveals the reliability of the software.
- Realization of Cinnamomun camphora essential oil extraction from a distance "" that, after the survey of the optimal parameters, gives a very good yield compared to the one found in the literature [11].

V. Conclusion

Visual Basic 6 is not an object oriented language but it presents this approach [12]. With the possibility to manipulate some objects, we could optimize our application while reusing some codes. It provides more performance to our application.

We also developed a convivial interface that satisfies the user's needs: facilitation of use, remote control and camera, all dimension systems, evolutionary interface. It provides an increased functional capacity as well as an increased portability.

References

[1]BERMAN, R.T. "Using C++ to write Automation Controller" Journal for Association of Laboratory Automation 12 (2007), 12-16. Original Research Article.

[2]Ali Shahrokni, Robert Feldt "A systematic review of software robustness", Information and Software Technology, Volume 55, Issue 1, January 2013, Pages 1–17. Original Research Article.

[3]En, Lyytinen, K., Loucopoulos, P., Mylopoulos, J. et Robinson, W. (dir.). « Conception Ingénierie des exigences: une perspective décennale » Springer-Verlag, 2009, pp 103-136.Original Research Article.

[4]. BehshidBehkamala, , , Mohsen Kahanib., Mohammad KazemAkbaric, " Customizing ISO 9126 quality model for evaluation of B2B applications", Information and Software Technology, Volume 51, Issue 3, March 2009, Pages 599–609. Original Research Article.

[5]Ho-Won Jung, « Validating the external quality subcharacteristics of software products according to ISO/IEC 9126"Computer Standards & Interfaces, Volume 29, Issue 6, September 2007, Pages 653- 661. Original Research Article.

[6] « Human-machine interface in the automation world» http://www.automation.siemens.com/mcms/automation/fr/human-machine- interface/Pages/Default.aspx

[7] "User manual of the heating agitator IKA" Version PDF.

[8] C.N. Madu, C.H. Kuei, R.A. Jacob, "An empirical assessment of the influence of quality dimensions on organizational performance", International Journal of Production Research 34 (7) (1996) 1943–1962.

[9] « SIEMENS Automation Software » http://www.automation.siemens.com/mcms/automation/fr/automation-systems/automation- software/Pages/Default.aspx

[11] RAZAFINDRAZANAKOLONA Daniel, "Remote control of Ravintsara essential oil extraction through the InterNet-world-wide-web", DEA 2010, University of Fianarantsoa MADAGASCAR.

[12] Peter. Wright «Visual basic 6 »Edition Eyrolles (1998).

[13]. Alain April et Claude Laporte, « Assurance qualité logicielle 1: concepts de base » Lavoisier, 2011, page 387.

[14] Ralph, P., Wand, Y, « A proposal of formal definition of the concept of design».

[16] « MSDN: Microsoft Documentation Network ».